

QUALITY ASSURANCE

CURS 4

AGENDA

- **Automation**
- **Selenium**
- **Prezentare Maria Ciocoiu @ TRP**

CAND AM TESTAT SUFICIENT?

- Saptamana acesta...
- Cercetatori in domeniul securitatii au reusit sa penetreze DE LA DISTANTA autoturismele echipate cu conexiune la Internet si sa preia controlul unor functii critice precum:
 - Dezactivarea franelor
 - Franarea la alegere a unei singure roti
 - Oprirea motorului
 - Etc



AUTOMATION

CAND AUTOMATIZAREA DEVINE UN BENEFICIU?

- **Este posibil**
 - Exista un sistem pentru a fi automatizat
 - Exista un test plan
 - Exista test cases scrise pentru testarea manuala
 - Exista un tool pentru automated testing
- **Este benefic**
 - Valoarea testarii automate a fost calculata
 - Exista o nevoie concreta pentru automatizare (cerintele clientului sau starea proiectului)

AVANTAJE

- **Posibilitatea de a testa continuu/ciclic**
- **Secventa stricta de pasi executati**
- **Productivitate crescuta comparativ cu testarea manuala**
- **Automatizarea secventelor lungi**
- **Automatizarea proceselor care necesita operatiuni complexe**

DEZAVANTAJE

- **Crearea scripturilor necesita un effort substantial (nu intotdeauna justificat)**
- **Scripturile trebuie intretinute**
- **Testele executa secvente de actiuni programate, nu posedea nici un fel de inteligenta**

DEVELOPMENT CU TESTE MANUALE

- **Scris cod**
- **Upload intr-un “loc”**
- **Build**
- **Executarea manuala (in multe cazuri introducere date, pas cu pas, etc)**
- **Verificare loguri, database, valori variable, etc**
- **Daca nu functioneaza repeta**

DEVELOPMENT CU TESTE AUTOMATE

- **Unit testing**

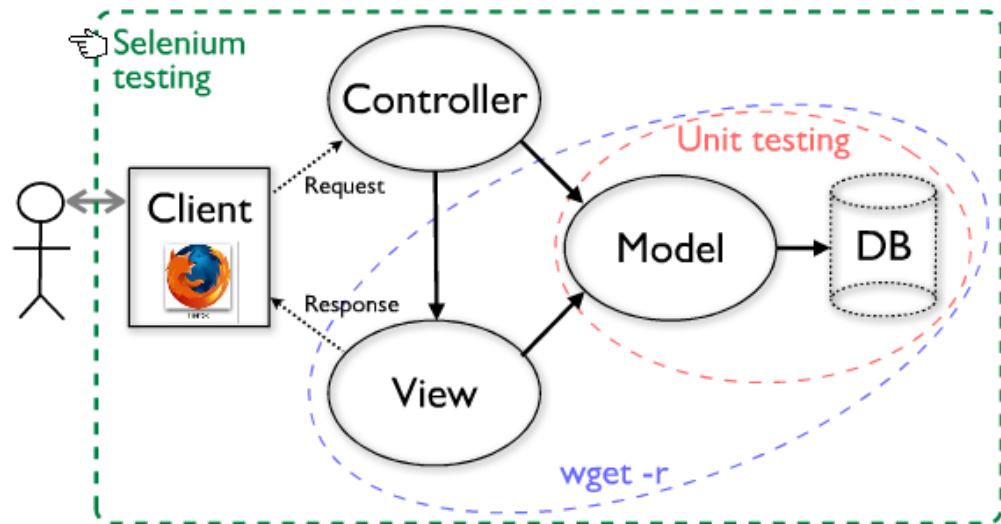
- Scris unul sau mai multe teste
- Build – testele nu trec
- Scris cod
- Build & executie
- Daca testul nu trece atunci fa schimbari
- Daca testul trece atunci repeta pentru urmatoarea metoda

- **Teste functionale**

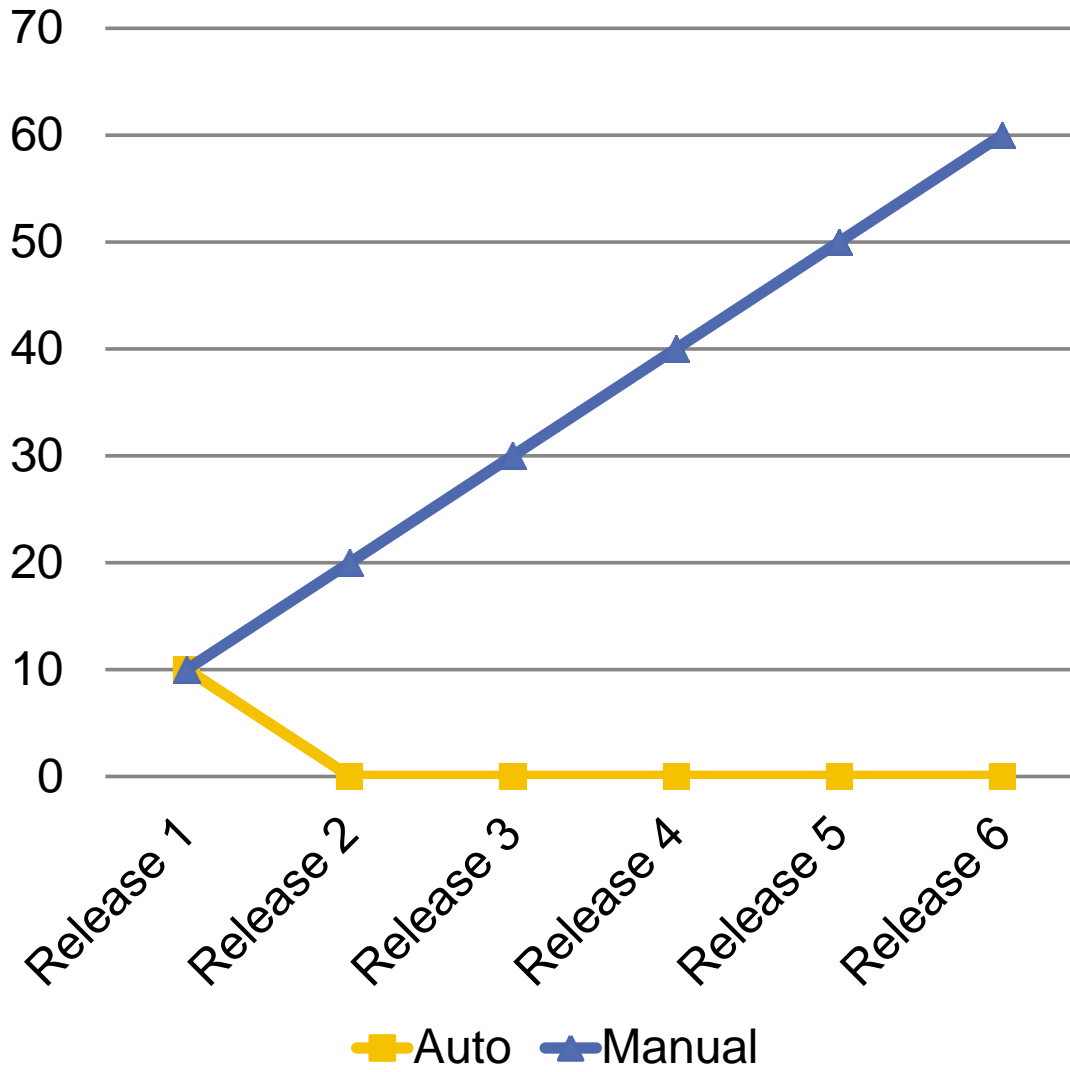
- Termina de scris cod (cu toate testele functionale trecute)
- Scrie un test functional
- Build & Executie
- Daca testul nu trece atunci fa schimbari
- Daca testul trece atunci repeta pentru urmatoarea metoda

BUILD IDEAL

- Ideal de executat aproximativ 10 minute
 - Checkout
 - Unit testing
 - Instalare baza date teste
 - Instalare server
 - Populare baza date test
 - Executie teste Selenium
 - Email cu rezultatele



EFORT SI COST



- 6 test cases
- Effort necesar pentru executie manuala => 10 min
- Effort necesar pentru automatizare => 10 min
- Effort necesar pentru executie automata => 1 min
- Numarul de iteratii => 5
- Timp total testare manuala => 50 min
- Timp total testare automata => 10 min

TESTAREA MANUALA E PLECTISITOARE

- **Testarea manuala e (cam) plectisitoare**
 - Nimeni nu viseaza sa completeze mereu aceleasi date in aceleasi formulare
 - Nu exista nimic nou de invatat la testarea manuala o data ce ai invatat cum se face
 - In general oamenii neglijeaza sa execute testele manuale
- **Testele automate sunt de fapt cod**
 - Sunt in general o provocare (intelectuala) mai mare (asta dupa ce au fost create test caseurile)
 - Trebuie avut in vedere designul si reutilizabilitatea
 - Necesita abilitati analitice
 - O contributie care e utilizabila si in viitor

REUTILIZABILITATE

- **Manual**

- Aceiasi cantitate de efort este necesara de fiecare data
- Testarea manuala nu poate fi reutilizata

- **Automation**

- Testele automate pot fi reutilizate (adica rulate automat de cate ori vrem)
- Foarte utile impreuna cu CI
- O data scris un test automat devine parte integranta din cod base

VIZIBILITATE

- **Manual**
 - Testele manuale ofera o vizibilitate redusa si trebuie mereu repetate de toti participantii in proiect (Tech Lead, Management, etc)
 - Doar programatorul care testeaza codul poate vedea rezultatele
- **Automation**
 - Oricine in proiect are acces instant la rezultatele testelor fara a le repeta
 - Nu necesita effort aditional pentru a verifica functionarea

IMPLEMENTARE

- **Release 1 – freeze/design/implement**
 - Code base care include unit testing si functional testing
 - Un server de CI
 - Framework pentru unit testing (xUnit)
 - Tool-uri pentru testare functionala (Selenium)
- **Atribuirea responsabilitatii**
 - Fiecare developer trebuie sa scrie unit testuri
 - Cineva care sa fie responsabil pentru executarea lor automata in procesul de CI

TOOLS

- **QuickTestPro**
 - Windows
 - Firefox/IE/Netscape
 - Webservices, Win App, Web, SAP, Flex, etc
 - Cost mediu: 20,000\$
- **IBM Rational Functional Tester**
 - IE/Firefox
 - Linux/Windows
 - Web, .NET, Java, Ajax, Flex, etc
 - Cost mediu: 6,000\$
- **Test complete**
 - IE/Firefox
 - Windows
 - .NET, Java, etc
 - Cost mediu: 2,000\$ / user
- **Selenium**
 - Firefox, Opera, IE
 - Windows/Unix/Mac
 - Web
 - Cost mediu: GRATIS

SELENIUM

- **Un framework de testare automata pentru aplicatii web**
- **Record/playback pentru crearea testelor fara a invata un limbaj**
- **Ofera un DSL (domain specific language) pentru a scrie teste in limbaje ca .NET, Ruby, PHP, etc**
- **Executia este posibila in multe browsere moderne**

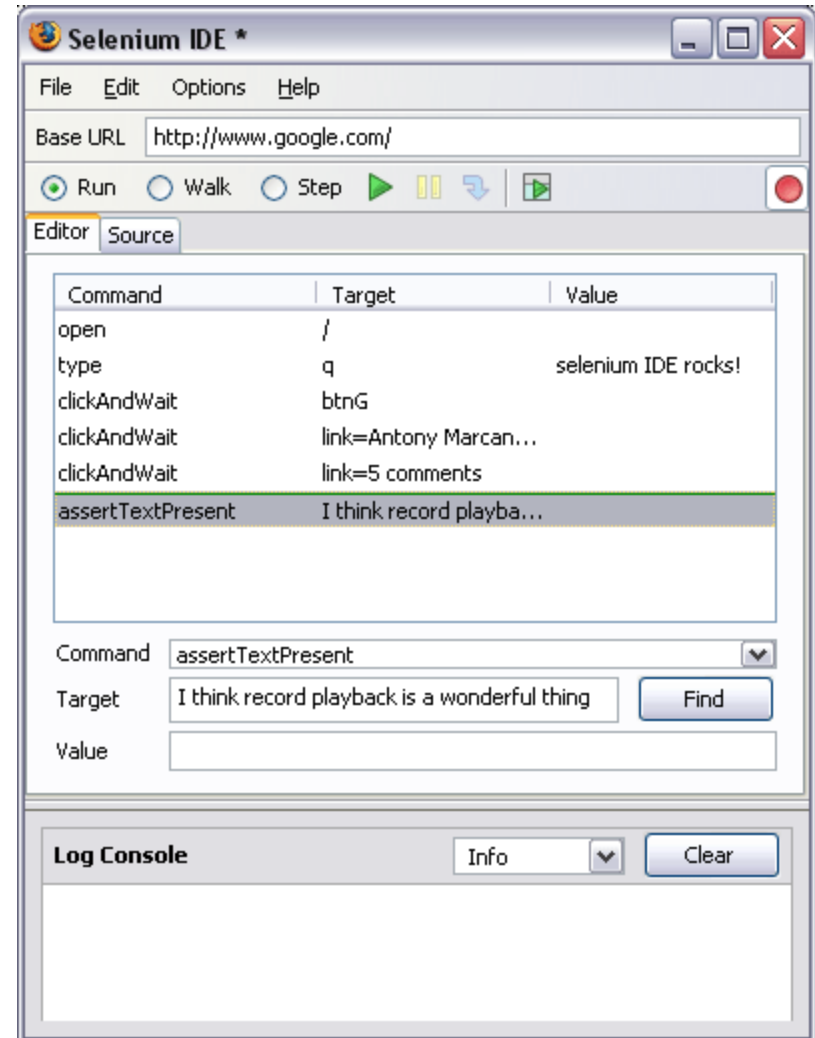
COMPONENTE

- **Selenium IDE**

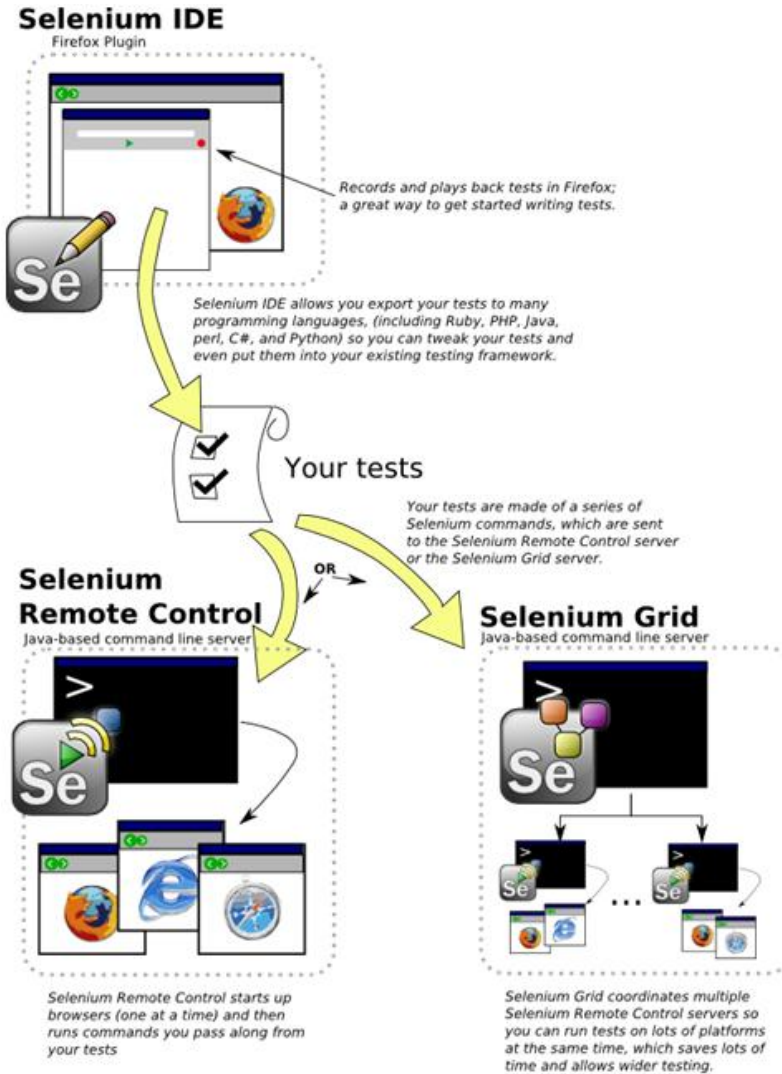
- Extensie Firefox pentru inregistrarea, editare si depanarea testelor
- Scripturile pot fi inregistrate sau editate manual
- Functionalitati:
 - Record and playback
 - Selectarea campurilor prin ID sau cu Xpath
 - Autocomplete pentru cele mai comune comenzi
 - Debug & Breakpoints
 - Save as... in limbaje cum ar fi Ruby, etc

- **Selenium Remote Control**

- Un server scris in Java ce accepta comenzi pentru un browser transmise prin HTTP
- Face posibila scrierea de teste automate pentru aplicatii web in orice limbaj, ceea ce permite integrarea mai buna a Selenium in proiecte



PASI



- Record/playback
- Export in limbajul preferat
- Executie fie folosind Selenium RC fie distribuind incarcarea pe mai multe servere folosind Selenium Grid

COMENZI

- **Un set de comenzi (un limbaj pentru testare) pentru a testa orice aplicatie web**
 - Existenta unor elemente de UI
 - Continut specific
 - Linkuri care nu functioneaza
 - Campuri de intrare
 - Selectarea de optiuni
 - Transmitteresa de formuri
 - Ajax
 - etc
- **3 Tipuri de comenzi: Actions, Accessors and Assertions.**
- **Actions sunt comenzi care in general manipuleaza starea aplicatiei**
 - “click link”, “select option”
 - Fail = executia testului se opreste
 - ..AndWait inseamna ca are loc un post-back si Selenium trebuie sa astepte executia
- **Accessors examineaza starea variabelor si folosite pentru Assert**
- **Assertions se asemana cu Accessors, dar verifica ca starea sistemului se conformeaza cu ceea ce este asteptat. Ex. “titlul paginii este X” sau “checkbox-ul este selectat”.**
 - 3 moduri: “assert”, “verify”, and ” waitFor”.
 - Ex. “assertText”, “verifyText” and “waitForText”.
 - “assert” care nu trece opreste un test
 - “verify” care nu trece va permite executia dar scrie in log eroarea
 - Ex. Cu “assert” verific ca sunt pe pagina corecta, cu “verify” verific valorile campurilor
- **“waitFor” asteapta indeplinirea unei conditii (util in testarea aplicatiilor cu AJAX)**

COMENZI UZUALE

- **click/clickAndWait** – realizeaza un click si optional astepta un rezultat
- **verifyTitle/assertTitle** – verifica titlul paginii
- **verifyTextPresent** – verifica prezenta unui text in pagina
- **verifyElementPresent** – verifica prezenta unui element de UI in pagina
- **verifyTable** verifica existenta unui table
- **waitForPageToLoad** opreste executia pana la incarcarea paginii.
Apelat automat atunci cand apelam **clickAndWait**

EXEMPLU (SITE LICITATI)

```
# User is not logged in. Searches for auction and chooses it in the listing
selenium.type "search_query", "selenium"
selenium.click "commit"
selenium.wait_for_page_to_load "30000"
selenium.click "link=Selenium test auction, single lot, normal+buy now"
selenium.wait_for_page_to_load "30000"
# User places a bid of 12 pounds (below the reserve price)
selenium.type "bid_max_price_vatted", "12"
selenium.click "bid_submit"
selenium.wait_for_page_to_load "30000"
# User logs in as Adalbert
selenium.type "login", "adalbert"
selenium.type "password", "asdf"
selenium.click "commit"
selenium.wait_for_page_to_load "30000"
# Bid confirmation page
selenium.click "bid_submit"
selenium.wait_for_page_to_load "30000"
# Redirected to auction page - check Adalbert is losing
assert selenium.is_text_present("You are losing your bid on this auction")
assert selenium.is_text_present("below the seller's reserve price")
```

SELENIUM PUNCTE TARI/SLABE

- **Tari**
 - Testare DOM simpla si puternica
 - Teste stabile bune pentru CI
 - Apt pentru testarea aplicatiilor AJAX
 - Testeaza scenarii tipice din aplicatie
 - Gratuit
- **Slabe**
 - Diagnoza erorilor
 - Schimbari valide de UI strica testele
 - Interactiunea cu Flash
 - Depinde de state

INVITAT SAPTAMANA VIITOARE

- **Gabriel Enea, Code40**
- **Senior developer Maxcode**
- **Microsoft Romania Community Influencer**
- **Co-fondatorul joobs.ro**
 - cel mai important site de joburi pentru IT din Romania
- **Prezentare: QA cu ASP.NET MVC**

INTREBARI

RASPUNSURI

PAUZA